Understanding /proc Precision Limits and the Need for Hardware Counters

1. Nature of /proc

The /proc filesystem (procfs) is a virtual interface exposing kernel data as human-readable text. It provides snapshots of kernel state variables at the time of reading rather than continuous data streams. Each file (e.g., /proc/stat, /proc/meminfo) is generated dynamically when accessed, meaning values are collected and formatted on-demand by the kernel.

2. Data Source and Update Mechanism

Values in /proc are derived from kernel accounting structures such as kernel_cpustat. These structures update according to the scheduler tick frequency (HZ): typically 100, 250, or 1000 Hz depending on kernel configuration, corresponding to update intervals of 10 ms, 4 ms, and 1 ms respectively.

3. Why /proc Cannot Be More Precise

- Kernel updates process and CPU counters only at scheduler tick boundaries (1–10 ms resolution).
- Reading /proc faster than this yields the same data repeatedly until the next tick.
- Text formatting and user-kernel copying introduce additional microsecond-scale latency.
- Each read involves a syscall and context switch, making microsecond-level sampling impractical.
- /proc was never designed for profiling but for system monitoring and debugging.

4. Frequency vs. Resolution Clarification

CPU frequency (in GHz) defines hardware execution speed, while kernel tick frequency (HZ) defines how often system accounting data updates. /proc reflects the latter, providing cumulative CPU time updated per tick, not continuous fine-grained measurements.

5. Why Hardware Counters Are Needed for High Precision

- perf / perf_event_open: access hardware performance counters (µs–ns resolution).
- ftrace / trace-cmd: capture kernel and scheduling events below 1 µs resolution.
- eBPF / bcc: run in-kernel aggregation with minimal user-space overhead.
- clock_gettime(CLOCK_MONOTONIC_RAW): obtain precise timestamps directly from hardware timers.

6. Key Observations

- /proc values are limited by kernel tick frequency (1–10 ms).
- Faster polling does not yield new data.
- /proc is text-based and aggregated, introducing additional delay.
- Excessive reading distorts timing and adds system load.

7. Final Conclusion

/proc provides coarse-grained, human-readable snapshots of system state, limited by kernel tick rate. Even on GHz-scale CPUs, accounting data updates only every few milliseconds. Reading /proc more frequently increases overhead without improving temporal precision. For microsecond or nanosecond measurements, hardware performance counters, kernel tracing, or high-resolution timers are required.

Essence in One Line:

/proc tells you what the kernel knew last tick — hardware counters tell you what's happening this microsecond.